

PROGRAMMING KEYBOARD status is a mixed bag. Bally still seems to have a July-August date for the appearance of the Keyboard. But there is an internal question now going on at Bally that asks if it might not be better to have a \$300. Keyboard with lesser capability (but expandable). The marketing surveys they have been running have indicated some kind of resistance (understandable) to a \$600+ unit. They have worked up a piece of hardware as a mockup to use internally for evaluation. But the decision (300, 600, maybe both?) must come soon if a 12-16 week production span is needed after go-ahead to stay within the 3rd Quarter window they had set up last year for availability.

HACKER'S MANUAL has appeared. I understand that it is being distributed by some dealers as an addendum to the regular manual. If your dealer does not have a copy of this 18-page document for you, I can make a copy and ship it out for \$2. Most of the data has already been included in the various issues of the ARCADIAN as our fellow subscribers have discovered them on their own. The 'new' material includes: some words on the I/O ports, a few words on the light pen interface, a block diagram of the sound synthesizer and description, and considerable detail as to wiring changes in the cassette interface to allow the addition of a printer jack.

LATE DATA on product availability...

2005	Star Battle	19.95	out in February
2007	Pinball	24.95	
3003	Grand Prix/Demolition Derby	19.95	
3004	Desert Fox/Drag Race	19.95	March
4004	Music	24.95	
5003	Backgammon/Checkers	19.95	

INTERACTIVE PROGRAMMING is being worked on by Jim Unroe. This is a scheme by which two machines can talk to each other via the cassette interfaces.

INTERCONNECTION to the S.D. Sales Z-80 CPU BOARD(kit \$139., P.O.Box 28810 Dallas 75228) is being explored by Pete Wishart up in Canada. He has developed a wiring scheme to to into the 50-pin connector on the back of the Arcade and wind up with an S-100-compatible interface. Still some bugs to be worked out.

DEALER in the Arkansas area is J.W.Taylor, 611 North 2nd, Cabot, 72023 who has an extensive supply, and I believe sends items postpaid.

LETTERHEAD of this issue was donated by Herb Weintraub. It is an interesting idea...

MENU can be called up by the following, donated by Martin Nason:

10 INPUT K

20 CALL K insert 3174

The menu will appear, and function fully(don't use the BASIC overlay card) but why does it not work if you just CALL 3174?

arcadian

ONBOARD CALCULATOR was very briefly mentioned in January. Here is some data on this feature. With this routine, it is possible to perform the four arithmetic functions with decimals, and use numbers much bigger than the Tiny BASIC limitation of 32767. But it takes up a lot of space. The operation is listed as \$ N @ (A), @ (B), @ (C) where N is the desired function + - * *

A is an input address, B is an input address, and C is the answer address. Each address is the beginning location of an 18-consecutive string, so that we could have A extending from 0 to 17, 18 to 35, 36 to 53, etc. B and C are similar. Within each of these sets, the decimal point is located at the near-center, the sign of the number is at the end, adjacent to an overflow indicator. Here is an illustration:

Sign: { + }	-----	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-------------	-------	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

overflow if #0 ----- decimal

Each digit of each input must be loaded independently, as well as its sign.

As an example, let us multiply 374.2913 by 96.7 to get 36193.96871:
Note the location of the decimal point and work from there-

Load the first input:

10 @ (10)=3; @ (9)=7; @ (8)=4; @ (7)=2; @ (6)=9; @ (5)=1; @ (4)=3

Load the second input

20 @ (27)=9; @ (26)=6; @ (25)=7

The registers will look like this:

@	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
@	0	0	0	0	0	0	0	3	7	4	2	9	1	3	0	0	0	0
@	11	10	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
@	0	0	0	0	0	0	0	0	9	6	7	0	0	0	0	0	0	0

List the operation:

30 \$*@ (0), @ (18), @ (36) ;RUN

The answer register looks like this:

@	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
@	0	0	0	0	0	0	0	0	3	6	1	9	3	9	6	8	7	1

and to recover it, include

40 FOR A = 53 TO 36 STEP -1

50 TV=@(A)

60 NEXT A

Which will yield 000003619396871000

This technique will suppress the leading zeros - adjust the values in lines 70 and 80 to fit the location of your answer address:

60 Z=1

70 IF @ (53)="8" PRINT "-" . If answer is a negative

80 FOR B=52 TO 44 STEP -1

90 IF @ (B)="0" IF Z GOTO 120

100 Z=0

110 TV=@(B)

120 NEXT B

The locations A,B, and C can be mixed up, or set equal to each other, or use other locations for memory, saving them for later use.

FORMATTING (PRINT #N) The following is a contribution from Tom Wood with some of my added comments and example. " A PRINT statement containing a #N value is most interesting. Apparently the value for N following the # sets the size of a 'window' to be left on the screen for each variable in the statement. Variables will be printed right-justified within that window.

A = 34; B= 973; C = -88; PRINT #4,A,B,".",C yields the following -

		3	4		9	7	3	.		-	8	8
--	--	---	---	--	---	---	---	---	--	---	---	---

We created a window of 4 character spaces wide for each variable on the PRINT line, noting that . is not a variable. The window is effective for the entire PRINT line, or until there is another #N " - Wood. This gives you the capability to create tabulated columns across the screen. To get this:

Program this:

```
10 A=391;B=7721;C=271;D=8143;  
      E=814;F=6392  
20 PRINT" JOHN DOE",#6,A,"-",#4,B  
30 PRINT" BILL WILSON",#6,C,"-",#4,D  
40 PRINT" HARRY JONES",#6,E,"-",#4,F  
JOHN DOE 391-7721  
BILL WILSON 271-8143  
HARRY JONES 814-6392
```

With the Onboard Calculator routine giving decimal calculation, you can start setting up material necessary for payroll accounting with answers in nice neat columns. Has anyone done any business programs?

PROGRAMS HERE, contributed by subscribers, include such games as CHECKERS, STRATEGY FOOTBALL, SLOT MACHINE, BALLY TREK, etc., and which are quite lengthy. I really haven't had time to give them a good scrubbing, but plan on doing so next month, and have them available for subscribers. I finally received a box of C-10 tapes from Microsystems, so now I can get organized.

PROGRAMS DIRECT from subscribers:

- o Bob Weber 6594 Swartout Rd. Algonac MI 48001 has the following available at \$2 plus a C-30 tape:

SUB SEARCH	ALIEN PATROL	CALENDAR
SLOT MACHINE	CONCENTRATION	TIC TAC TOE
FLIGHT SIMULATOR	HANGMAN	MATH QUIZ
OTHELLO	MASTERMIND	SPACE CHASE
o Ron Schwenk 6988 Lincoln Creek Circle, Carmichael CA	95608	
MASTERMIND	ONE CHECK	BATNUM
o Bob Strand 10665 E. Foix Ave. Norwalk CA 90650	\$7 for the lot...	
STAR BATTLES	4 DIGIT GUESS	REMEMBER
ANGLE GAME	SLOT MACHINE	NUMBER WAR
LUNAR LANDER(enhanced/expanded)		
o George Hale P.O.Box 186 Lee's Summit MO 64063	ashoot-it-down type	
of game for two that he calls SONIC SATELLITE. This will be available as a		
listing for \$4., as a cassette tape @ \$8.50, or loaded on your tape @ \$6.50.		
George will be selling Bally-oriented goods through Applications Programming		
Enterprise.		

FOR SALE Bally ARCADE with BASIC, CLOWNS, and BASEBALL, \$275. W.KIM,
776 Via Catalina, Burbank, CA 91504 (213-767-3963)

arcadian

<u>MEMORY MAP</u>	Decimal	Hexadecimal
On Board ROM	0- 8191	0-1FFF
Bally BASIC ROM	8192-12287	2000-2FFF
Screen Memory Area	16384-20479	4000-4FFF
Bally BASIC Graphics/Program Area		
	16384-19983	4000-4E10
Bally BASIC Scratchpad Memory Area		
	20000-20463	4E20-4FEE
Tape Input Buffer	20002-20049	4E22-4E51
Variables begin at	20078	4E6E
Line Input Buffer (104 characters)	20180-20283	4ED4-4F3B
Stack Area	20284-20462	4F3C-4FEE
Text Area	24576-22777	A000-A707
Note Lookup Table	12046	2FOE for CR(13 ₁₀)

The above was extracted from the Hacker's Manual.

SPACE SAVER has been located by Bob Weber - If a PRINT "X" is not followed by another command, the final " is not needed. "A byte saved is a byte available for another statement."

ANOTHER DIVISION ROUTINE that prints a decimal answer has been developed by Pete Bowman. This one is a bit laborious as you have to enter a @() for each decimal wanted, in line 80.

```
10 PRINT "X- Y = Z"      50 FOR W = 1 TO N  (where N is the number of
20 INPUT "X=? " X          decimal digits desired)
30 INPUT "Y=? " Y          60 @W) = (RM*10) ÷ Y
40 Q = X ÷ Y          70 NEXT W
                      80 PRINT "Z=", #1,Q,".",@1,@2,@3,...@N)
```

NOTE TIME has been noted by many to control speed of operations to some extent. Setting it =0 makes things operate the fastest. Negative numbers yield very slow results. You can also go back and forth to tape faster with :PRINT;NT=1;LIST Using NT=0 here doesn't always work.

PROGRAMS INCLUDED this month are short enough to put on a page. The form that I used was provided by Chuck Thomka, 1228 W.222 St. Torrance CA 90502. It is a handy way to keep things in order. Program listing should be reviewed as a training aid, to help in your own understanding.

PROGRAM NAME

Page of

Line #	Statement(s)	Comments
1	SIMON	
2	BY BRETT BILBRY	
3	AND JOE BORRELLO	
10	CLEAR; &(0)=7; &(1)=7; &(2)=0 &(3)=0; &(9)=30; NT=0; CX=47 CY=-20; PRINT "SCORE:" ; NT=30	
20	BOX 0, 20, 22, 22, 1; BOX 0, 20, 20, 20, 2	
30	BOX 0, -20, 22, 22, 1; BOX 0, -20, 20, 20, 2	
40	BOX 20, 0, 22, 22, 1; BOX 20, 0, 20, 20, 2	
50	BOX -20, 0, 22, 22, 1; BOX -20, 0, 20, 20, 2	
60	A=A+1; @ (A)=RN(), 4)	
70	FOR X=1 TO A; IF @ (X)=1, GOSUB 130; GOTO 110	
80	IF @ (X)=2, GOSUB 2000; GOTO 110	
90	IF @ (X)=3, GOSUB 3000; GOTO 110	
100	GOSUB 4000; GOTO 110	
110	NEXT X; FC=0; FOR X=1 TO A, B=JX(1); C=JY(1); IF B=0, IF C=0 GOTO 120	
120	IF B≠0, D=(1+B)÷2+3 IF C≠0, D=(1-C)÷2+1	
130	IF D≠@ (X) NT=55; MU=33; MU=48; NT=3; FC=0; FOR X=1 TO 5000; NEXT X; GOTO 10	
140	GOSUB D+1000; GOTO 170	
150	FC=0; NEXT X	
160	CX=23; CY=-30; NT=4; PRINT A; NT=30	
170	FOR O=1 TO 500; NEXT O; GOTO 60	
180	MU=44; BOX 0, 20, 20, 20, 1; MU= 49; FOR O=1 TO 255-2+A; NEXT O ; BOX 0, 20, 20, 20, 2; RETURN	
190	FC=204; BOX 0, -20, 20, 20, 2; MU=51; FOR O=1 TO 255-2+A; NEXT O; BOX 0, -20, 20, 20, 2; RETURN	
200	FC=252; BOX -20, 0, 20, 20, 2; MU=53; FOR O=1 TO 255-2+A; NEXT O; BOX 0, -20, 0, 20, 2; RETURN	
210	FC=134; BOX 20, 0, 20, 20, 2; MU=55; FOR O=1 TO 255-2+A; NEXT O; BOX 20, 0, 20, 20, 2; RETURN	
220		
230		
240		
250		
260		
270		
280		
290		
300		
310		
320		
330		
340		
350		
360		
370		
380		
390		
400		
410		
420		
430		
440		
450		
460		
470		
480		
490		
500		
510		
520		
530		
540		
550		
560		
570		
580		
590		
600		
610		
620		
630		
640		
650		
660		
670		
680		
690		
700		
710		
720		
730		
740		
750		
760		
770		
780		
790		
800		
810		
820		
830		
840		
850		
860		
870		
880		
890		
900		
910		
920		
930		
940		
950		
960		
970		
980		
990		
1000		
1010		
1020		
1030		
1040		
1050		
1060		
1070		
1080		
1090		
1100		
1110		
1120		
1130		
1140		
1150		
1160		
1170		
1180		
1190		
1200		
1210		
1220		
1230		
1240		
1250		
1260		
1270		
1280		
1290		
1300		
1310		
1320		
1330		
1340		
1350		
1360		
1370		
1380		
1390		
1400		
1410		
1420		
1430		
1440		
1450		
1460		
1470		
1480		
1490		
1500		
1510		
1520		
1530		
1540		
1550		
1560		
1570		
1580		
1590		
1600		
1610		
1620		
1630		
1640		
1650		
1660		
1670		
1680		
1690		
1700		
1710		
1720		
1730		
1740		
1750		
1760		
1770		
1780		
1790		
1800		
1810		
1820		
1830		
1840		
1850		
1860		
1870		
1880		
1890		
1900		
1910		
1920		
1930		
1940		
1950		
1960		
1970		
1980		
1990		
2000		
2010		
2020		
2030		
2040		
2050		
2060		
2070		
2080		
2090		
2100		
2110		
2120		
2130		
2140		
2150		
2160		
2170		
2180		
2190		
2200		
2210		
2220		
2230		
2240		
2250		
2260		
2270		
2280		
2290		
2300		
2310		
2320		
2330		
2340		
2350		
2360		
2370		
2380		
2390		
2400		
2410		
2420		
2430		
2440		
2450		
2460		
2470		
2480		
2490		
2500		
2510		
2520		
2530		
2540		
2550		
2560		
2570		
2580		
2590		
2600		
2610		
2620		
2630		
2640		
2650		
2660		
2670		
2680		
2690		
2700		
2710		
2720		
2730		
2740		
2750		
2760		
2770		
2780		
2790		
2800		
2810		
2820		
2830		
2840		
2850		
2860		
2870		
2880		
2890		
2900		
2910		
2920		
2930		
2940		
2950		
2960		
2970		
2980		
2990		
3000		
3010		
3020		
3030		
3040		
3050		
3060		
3070		
3080		
3090		
3100		
3110		
3120		
3130		
3140		
3150		
3160		
3170		
3180		
3190		
3200		
3210		
3220		
3230		
3240		
3250		
3260		
3270		
3280		
3290		
3300		
3310		
3320		
3330		
3340		
3350		
3360		
3370		
3380		
3390		
3400		
3410		
3420		
3430		
3440		
3450		
3460		
3470		
3480		
3490		
3500		
3510		
3520		
3530		
3540		
3550		
3560		
3570		
3580		
3590		
3600		
3610		
3620		
3630		
3640		
3650		
3660		
3670		
3680		
3690		
3700		
3710		
3720		
3730		
3740		
3750		
3760		
3770		
3780		
3790		
3800		
3810		
3820		
3830		
3840		
3850		
3860		
3870		
3880		
3890		
3900		
3910		
3920		
3930		
3940		
3950		
3960		
3970		
3980		
3990		
4000		
4010		
4020		
4030		
4040		
4050		
4060		
4070		
4080		
4090		
4100		
4110		
4120		
4130		
4140		
4150		
4160		
4170		
4180		
4190		
4200		
4210		
4220		
4230		
4240		
4250		
4260		
4270		
4280		
4290		
4300		
4310		
4320		
4330		
4340		
4350		
4360		
4370		
4380		
4390		
4400		
4410		
4420		
4430		
4440		
4450		
4460		
4470		
4480		
4490		
4500		
4510		
4520		
4530		
4540		
4550		
4560		
4570		
4580		
4590		
4600		
4610		
4620		
4630		
4640		
4650		
4660		
4670		
4680		
4690		
4700		
4710		
4720		
4730		
4740		
4750		
4760		
4770		
4780		
4790		
4800		
4810		
4820		
4830		
4840		
4850		
4860		
4870		
4880		
4890		
4900		

Line

Statement(s)

5 .CLOCK. BY J. COUSINS

6 .

7 .

1.0 CLEAR; BC=0; FC=22

2.0 INPUT "HOURS"; H

3.0 IF H>23 GOTO 20

4.0 INPUT "MINUTES"; M

5.0 IF M>59 GOTO 40

6.0 INPUT "SECONDS"; S

7.0 IF S>59 GOTO 60

8.0 CLEAR

9.0 FOR I=1 TO 70

10.0 NEXT I; NT=0

11.0 B=B+1

12.0 IF S<=59 GOTO 180

13.0 S=S+1

14.0 IF M<=59 GOTO 180

15.0 M=M+1

16.0 IF H<=23 GOTO 180

17.0 H=H+1

18.0 CX=-1.2; CY=2.0; A=15

19.0 IF H<13 A=A

20.0 A=A+65

21.0 TV=A; TV=7.7

22.0 CX=-3.0; CY=0; T=H

23.0 GO SUB 290

24.0 TV=58; T=M

25.0 GOSUB 290

26.0 TV=58; T=S

27.0 GOSUB 290

28.0 NT=1; MU=7.6; GOTO 90

29.0 A=ABS((T+1.0)); B=R.M

30.0 TV=A+48; TV=B+48

31.0 RETURN

Line

Statement(s)

5 . CONVERT HEX TO DECIMAL

6 . BY ERNIE SAMIS 3-1-79.

7 .

1.0 CLEAR

2.0 M=0; PRINT

3.0 FOR N=0 TO 3

4.0 PRINT "HEX #"

5.0 K=K.P

5.2 IF K<47 GOTO 200

5.4 IF K>70 GOTO 240

5.6 IF K>57 IF K<65 GOTO 290

6.0 TV=K; PRINT

7.0 N=N-48

8.0 IF K>57 N=N-7

9.0 IF A=0 IF N>7 M=-327673

M=M+(4995*(N-7))+N-73

M=M-4996; IF N>8 M=M-13

GOTO 150

10.0 IF A=4 IF N=8 GOTO 250

11.0 IF A=4 M=M+(4996*N)

12.0 IF A=1 M=M+(256*N)

13.0 IF A=2 M=M+(16*N)

14.0 IF A=3 M=M+(1*N)

15.0 NEXT A

16.0 PRINT; PRINT "THE DECIM

17.0 GOTO 240

240 PRINT; TV=K; PRINT "IS A,

5. INVALID HEX #"; GOTO 240

DO NOT ENTER A SPACE BETWEEN LINE #

AND STATEMENT THIS IS DONE BY THE UNIT

USE OF SHADED AREA IS FOR END OR

MORE LINES OF MULTI-LINE STATEMENTS

DO NOT ENTER A SPACE BETWEEN LINE #

AND STATEMENT THIS IS DONE BY THE UNIT

USE OF SHADED AREA IS FOR END OR

MORE LINES OF MULTI-LINE STATEMENTS

DO NOT ENTER A SPACE BETWEEN LINE #

AND STATEMENT THIS IS DONE BY THE UNIT

USE OF SHADED AREA IS FOR END OR

MORE LINES OF MULTI-LINE STATEMENTS

DO NOT ENTER A SPACE BETWEEN LINE #

AND STATEMENT THIS IS DONE BY THE UNIT

USE OF SHADED AREA IS FOR END OR

MORE LINES OF MULTI-LINE STATEMENTS

DO NOT ENTER A SPACE BETWEEN LINE #

AND STATEMENT THIS IS DONE BY THE UNIT

USE OF SHADED AREA IS FOR END OR

MORE LINES OF MULTI-LINE STATEMENTS

DO NOT ENTER A SPACE BETWEEN LINE #

AND STATEMENT THIS IS DONE BY THE UNIT

USE OF SHADED AREA IS FOR END OR

MORE LINES OF MULTI-LINE STATEMENTS

command name	function
box	draws a rectangle on the screen & has options for building picture prototype lists
*change	changes the values of an endpoint in a picture prototype list
circle	draws an ellipse on the screen & has options for building picture prototype lists
clear	clears the screen
*close	closes off an open picture prototype list
colors	chooses 4 colors of 256 for screen use
*compile	compiles code for speed
copy	makes a copy of a picture prototype with a new name
delete	deletes and reclaims storage of a named thing
display	causes a picture prototype to be exclusive or'ed onto the screen and be updated when necessary
*film	sets up filming mode for a Super 8 camera
*fetch	retrieves a given endpoint in a picture prototype list
get	gets a macro, array, picture prototype list, etc. from tape, disk, etc.
group	collects picture prototypes into a group which can be referenced with a single name. Transformations may be done to the group as a whole or to individual members.
help	prints commands and required argument types
ieee	provides interface to IEEE bus
input	used to input numbers, strings from terminal or passed argument lists
line	draws a vector & has options for building picture prototype lists
memory	gives a usage map of memory
move	attaches a picture prototype to two variables, devices, etc. so that when they change, the prototype is automatically erased and redrawn in the new position with options for "exclusive or" or "load/store" read and write to screen
*onerror	traps errors to a user's routines
*open	allocates storage and starts up a picture prototype list
*pattern	allows a pixel list to be directly built rather than snapped
play	interprets a string, array or picture prototype as a musical score to be played by the three-voice synthesizer
put	stores a macro, array, picture prototype list, etc. on tape, disk, etc.
rename	renames a named thing to a new name
*rotate	like move but the prototype is rotated
*scale	like move but the prototype is scaled
select	causes picture prototypes to be switched round-robin fashion on the screen
snap	takes a screen image in rectangular bounds and makes it into a movable picture prototype
sync	tells the system how much time to devote to interrupt-level updating versus command processing
*vip	allows a macro to be executed at interrupt level (stands for "very important program")

ZGRASS COMMANDS are listed here. These are some of the unique ones planned for the Keyboard's language. The machine I saw had a total of 66 commands. This page followed "page 36" of the article reproduced in ARCADIAN #2.

PROGRAM NAME

Line #	Statement(s)
1	. REVERSE
2	. BY. BRETT BILBRAY.
3	. AND MIKE TOTH.
10	N=9;CLEAR.
20	Q(I)=RND(9)
30	FOR K=2 TO N
40	Q(K)=RN(9)
50	FOR J=1 TO K-1
60	IF Q(K)=Q(J) GOTO 40
70	NEXT J; NEXT K
80	CY=25
90	PRINT "THE LIST IS"
100	T=4
110	GOSUB 200
120	CX=-2.4;CX=.4;Q=.4;NT=.4
130	O=KN(1)÷32+.6;IF TR(1)
	=160 TO 169
140	IF O#Q CX=-.4;PRINT O;Q=O
150	GOTO 130
160	I=I+1; NT=.3
170	FOR K=ITO 9#Z
180	Z=Q(K)
190	Q(K)=Q(K+1)
200	Q(K-K+1)=Z
210	NEXT K
220	GOSUB 200
230	FOR K=ITO 9;IF Q(K)≠K
	GOTO 120
240	NEXT K
250	CX=-.7.4
260	CY=-2.4;PRINT " YOU WON IN "
	;TV=T÷1.6+4.8;TV=T÷T+1.6+4.8;
	PRINT "MOVES"
270	FOR X=ITO 5#ODD;NEXT X;
	GOTO 10
280	CX=-.5;CY=.4
290	FOR B=ITO N;TV=4.8+Q(B);
	TV=32.3; NEXT B
300	RETURN

GAME INSTRUCTIONS These games were sent by Brett Bilbray who welcomes comments and suggestions at 14430 Barclay, Dearborn, MI 48126.

SIMON: One player, Hand Controller

The computer shows you a pattern that you have to repeat, using joy stick controls.

REVERSE: One player, Hand Controller

The object is to get 9 number in order (smallest at the left) that are initially in random order. Use the knob to identify the numbers to be moved, and the trigger to move them.

-38-

ARCADIAN

Robert Fabris, Boss
3626 Morrie Dr.
San Jose, CA 95127

FIRST CLASS